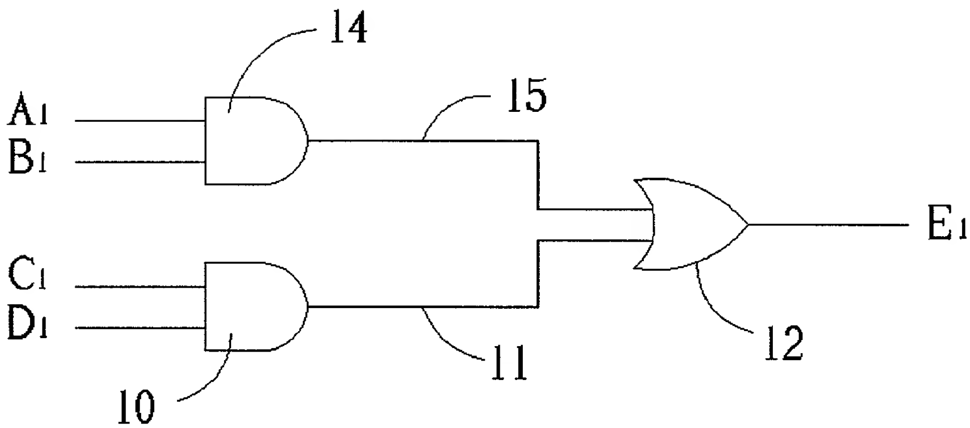

Fig. 1

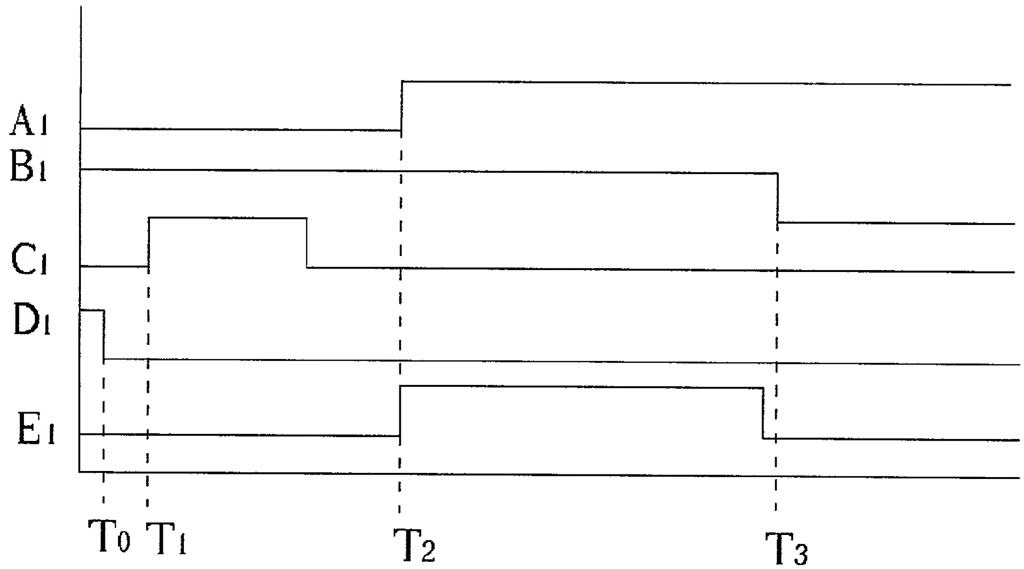

Fig. 2

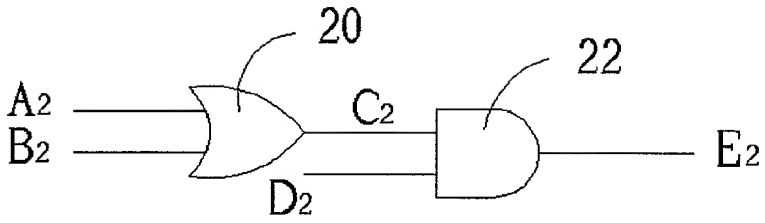

Fig. 3

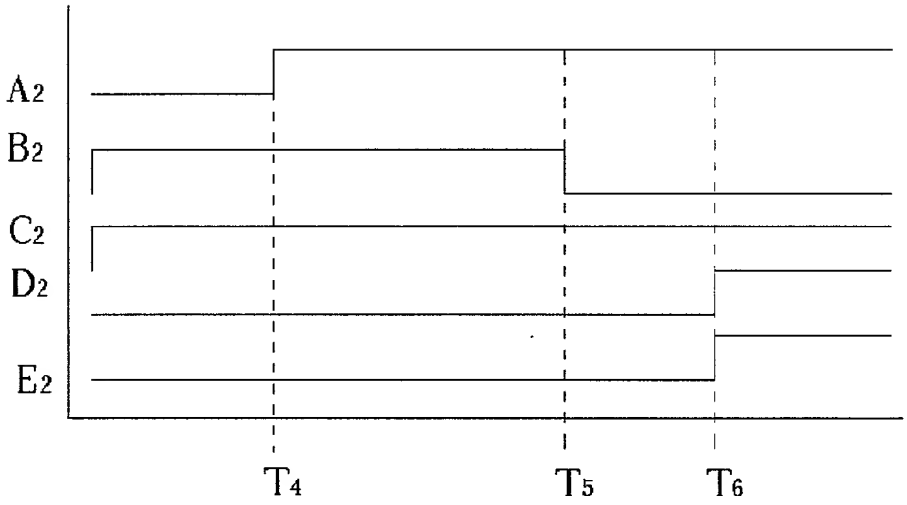

Fig. 4

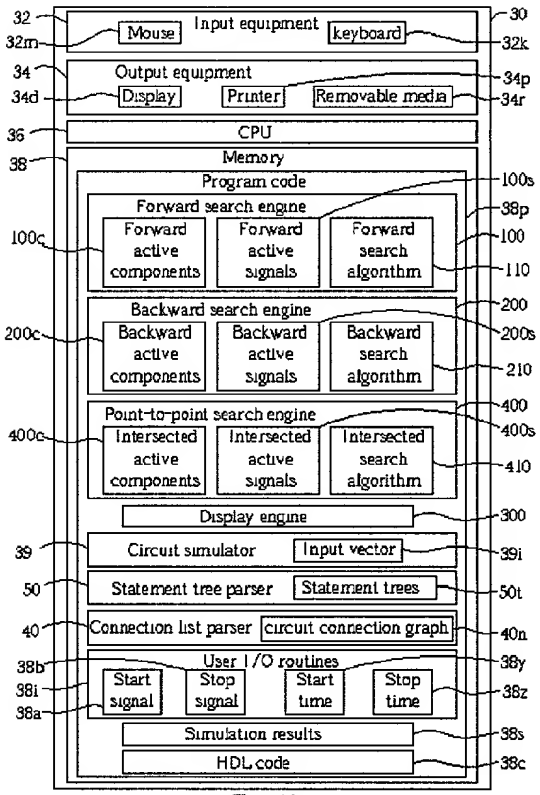

Fig. 5

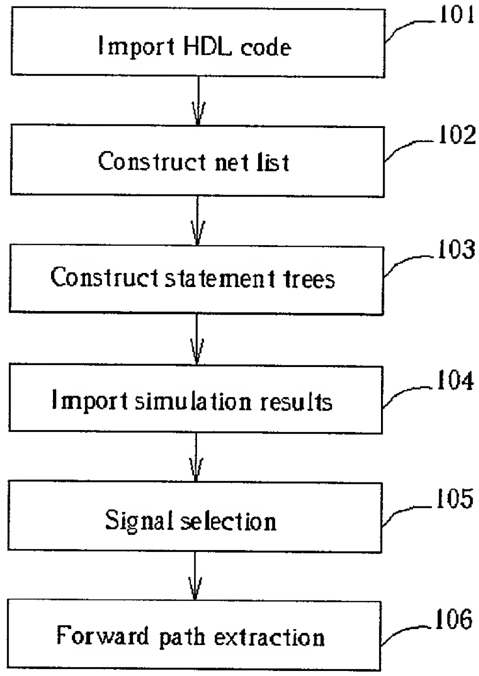

Fig. 6

```
    module
    example(clk,reset,controlc,controld,controle,ind,ine,ing,inh,ini,outf,outg,outh);
    input clk,reset,ind,ine,ing,inh,ini;
    input controlc,controld,controle;
    output outf,outg,outh;
    wire controlc,controld,controle;
    wire ind,ing,inh,ini;
    wire outf;
    reg outh,outg;
    wire sigc,sige,sigh;
    reg siga,sigb,sigd,sigf,sigg,sigi;
C7  assign outf = siga & sigb;
C1  assign sigc = ine | ing;
C3  assign sige = sigi & ini;
C8  assign sigh = ine & inh;

C9  always @(posedge clk or posedge reset)
    begin
            if (reset)
                    outh = 0;
            else if (controlc || controld)
                    outh = #1 sigh;
            else if (!controle)
                    outh = #1 ing;
    end

C11 always @(posedge clk or posedge reset)
    begin
            if (reset)
                    sigf = 0;
            else if (controld)
                    sigf = #1 ind | inh;
            else if (!controle)
                    sigf = #1 ing;
    end
```

Fig. 7A

```
C6   always @(posedge clk or posedge reset)
     begin
         if(reset)
                 outg = 0;
         else if (controlc)
                 outg = sige;
         else
                 outg = sigc;
     end
C2   always @(posedge clk or posedge reset)
     begin
         if(reset)
                 sigd = 0;
         else if (controlc || controld)
                 sigd = #1 ind;
         else if (!controle)
                 sigd = #1 sigc;
     end
C5   always @(posedge clk or posedge reset)
     begin
         if (reset)
                 siga = 0;
         else if (controlc)
                 siga = #1 sigd;
         else
                 siga = #150 sigf;
     end
C4   always @(posedge clk or posedge reset)
     begin
         if(reset)
                 sigb = 0;
         else if (controlc && !controld)
                 sigb = #1 sige;
         else if (controle)
                 sigb = #1 sigg;
     end
```

Fig. 7B

C10

```
always @(posedge clk or posedge reset)
begin
        if (reset)
                sigg = 0;
        else if (controlc && !controld)
                sigg =  #1 ini;
        else if (controle)
                sigg =  #1 inh;
end
```

C12

```
always @(posedge clk or posedge reset)
begin
        if (reset)
                sigi = 0;
        else
                sigi =  #1 inh;
end
endmodule
```

Fig. 7C

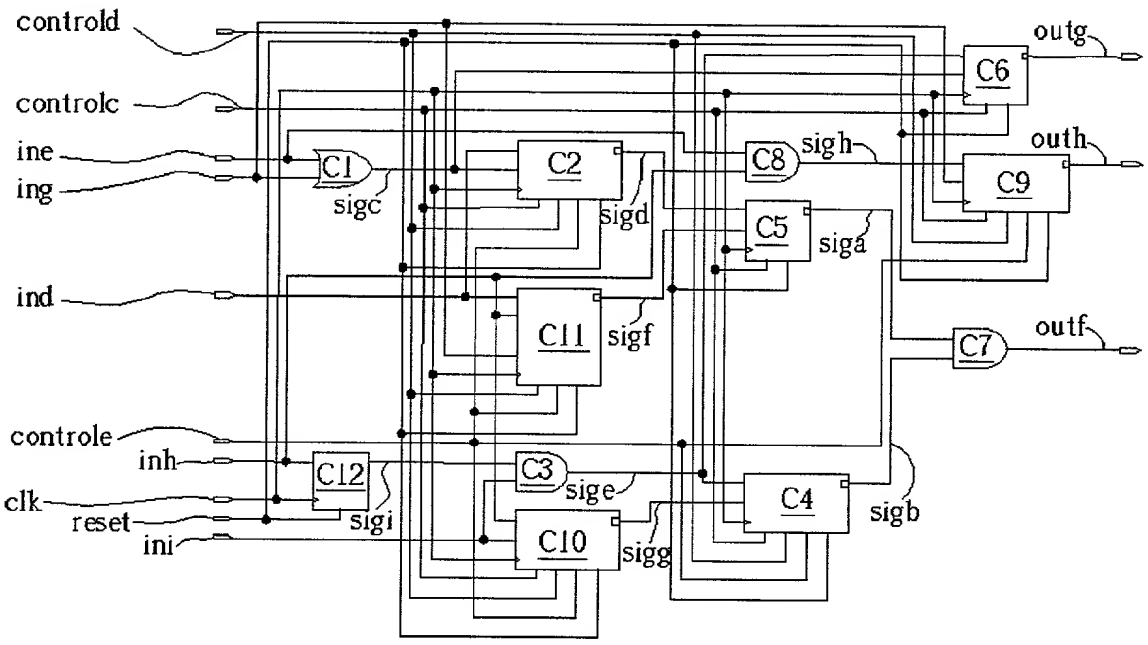

Fig. 8

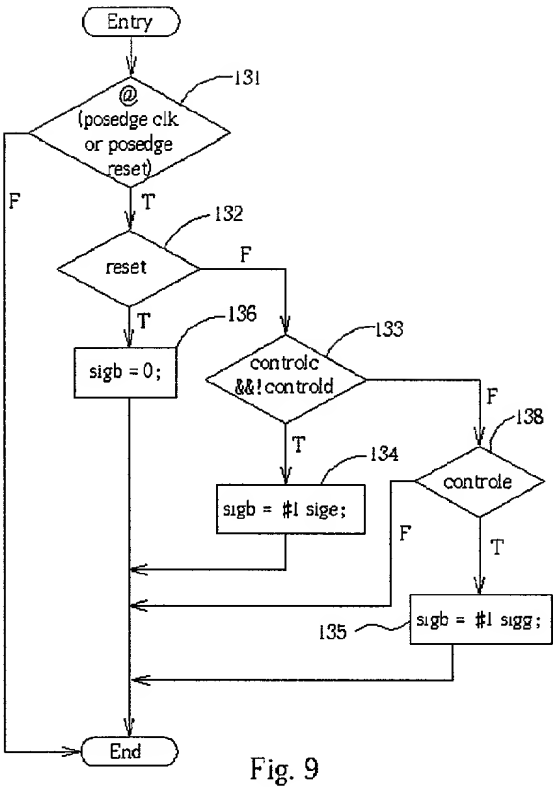

Fig. 9

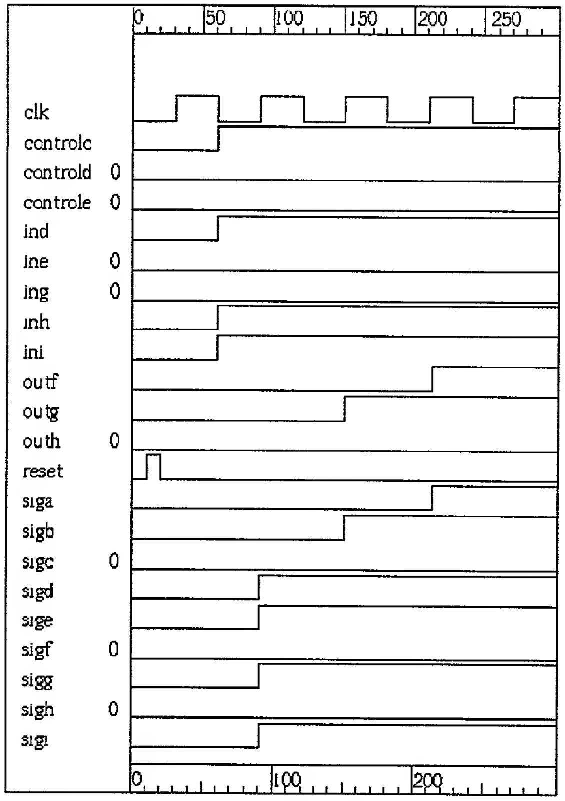

Fig. 10

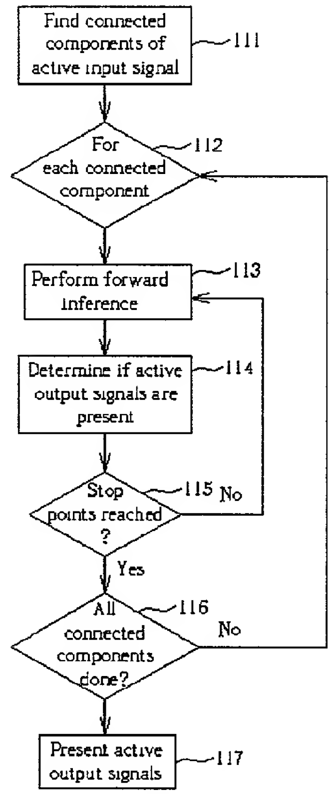

Fig. 11A

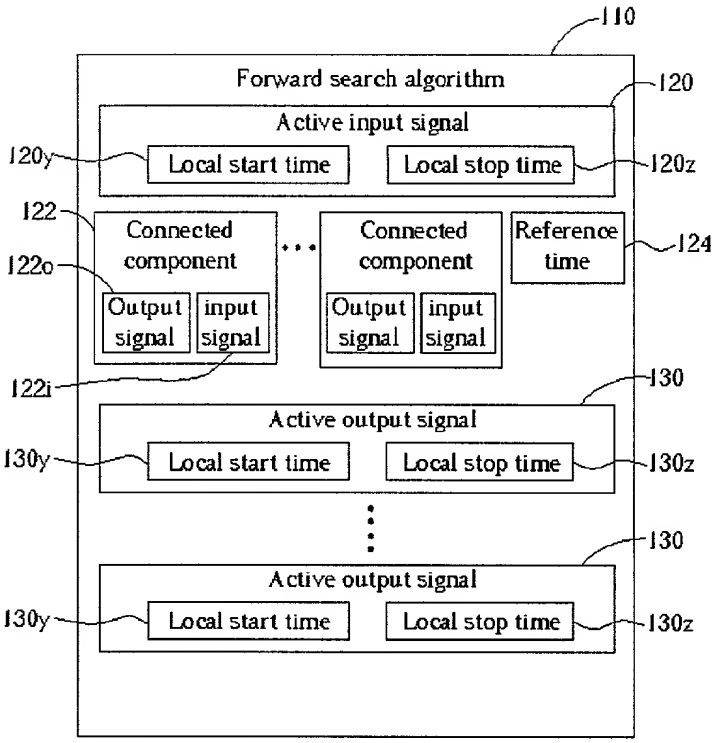

Fig. 11B

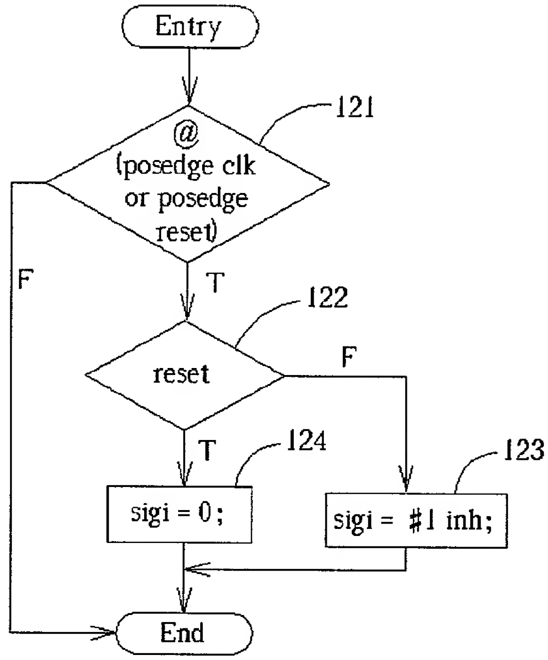

Fig. 12

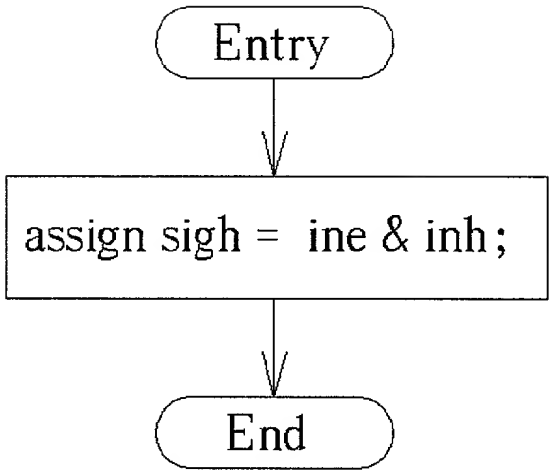

Fig. 13

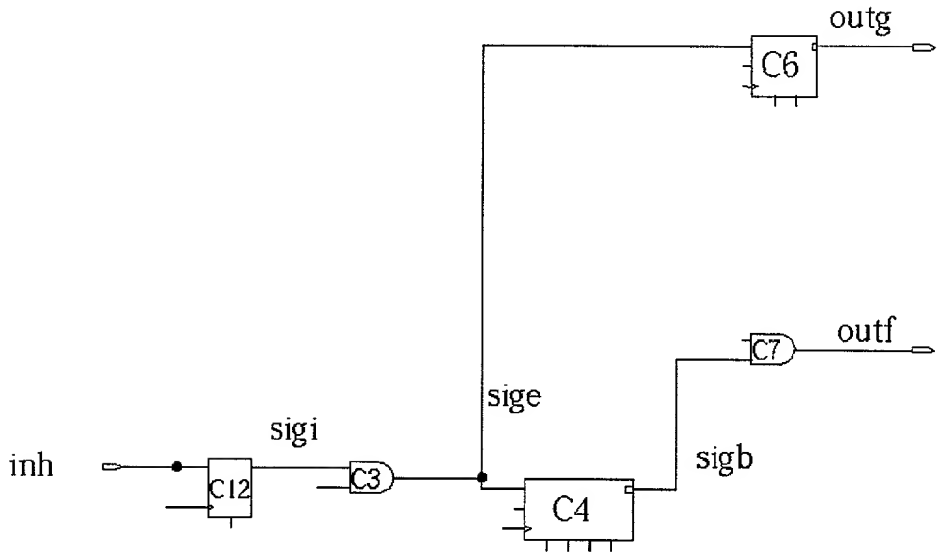

Fig. 14A

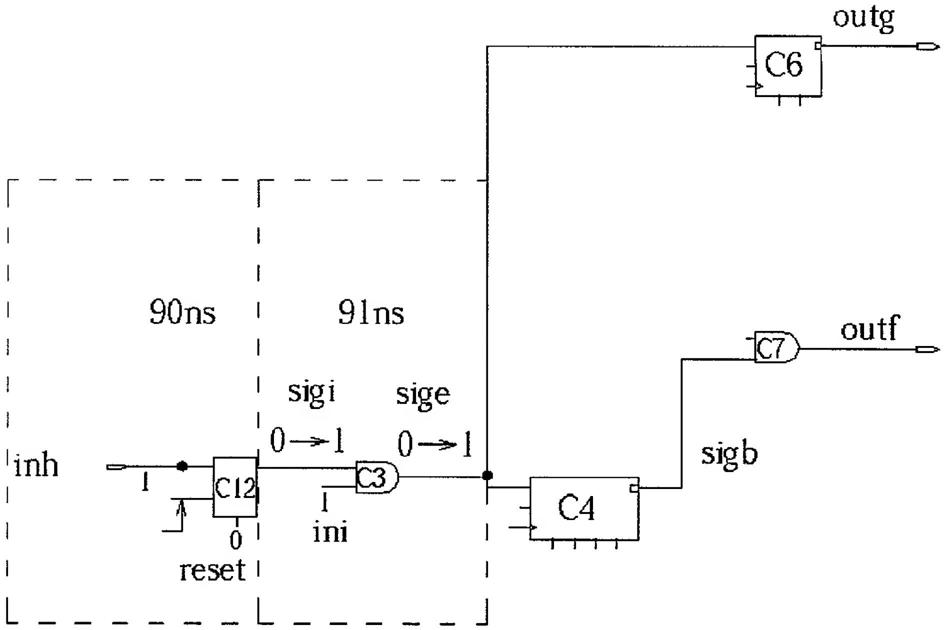

Fig. 14B

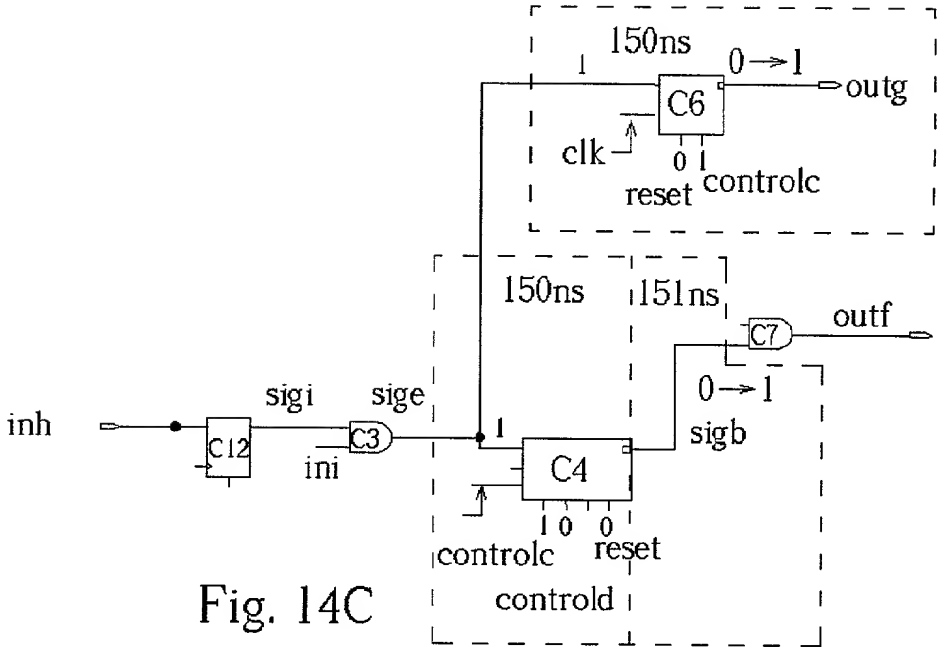

Fig. 14C

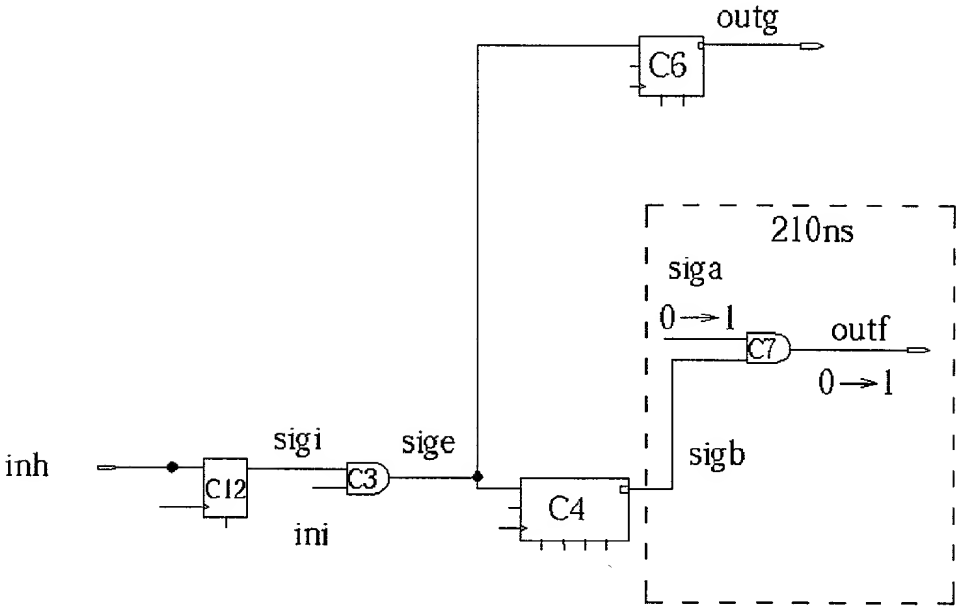

Fig. 14D

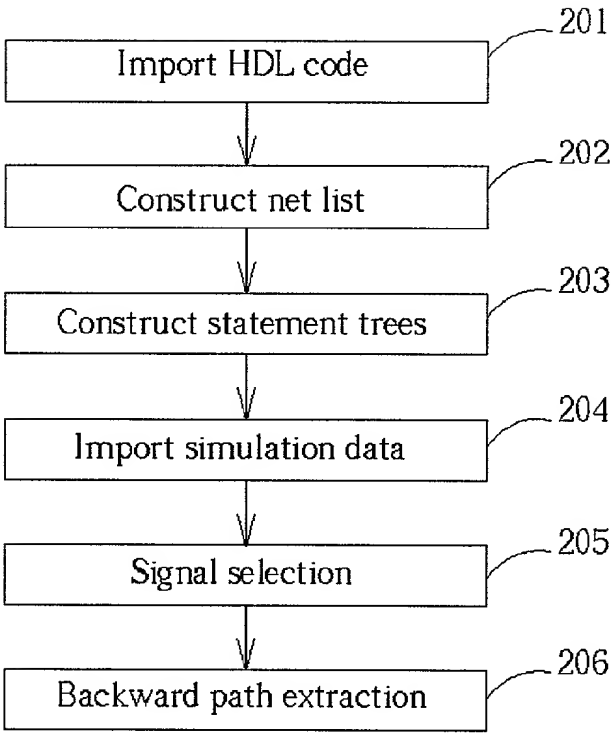

Fig. 15

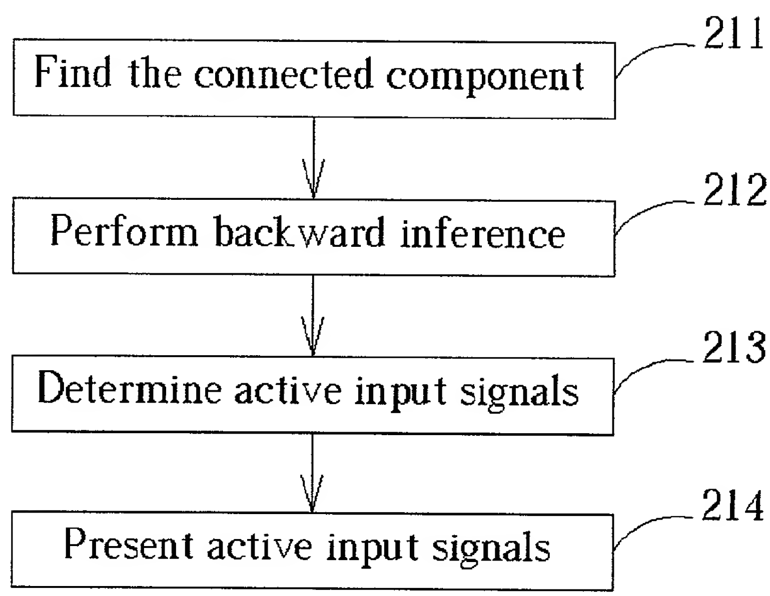

Fig. 16A

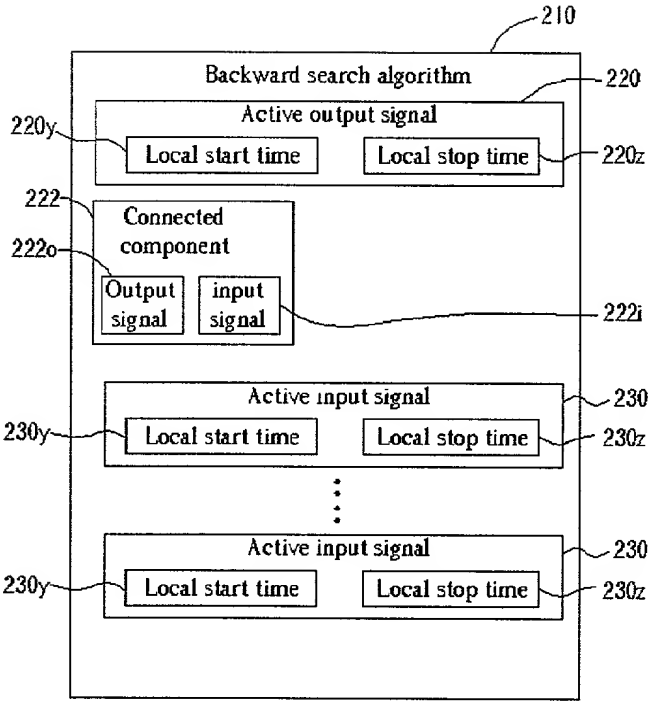

Fig. 16B

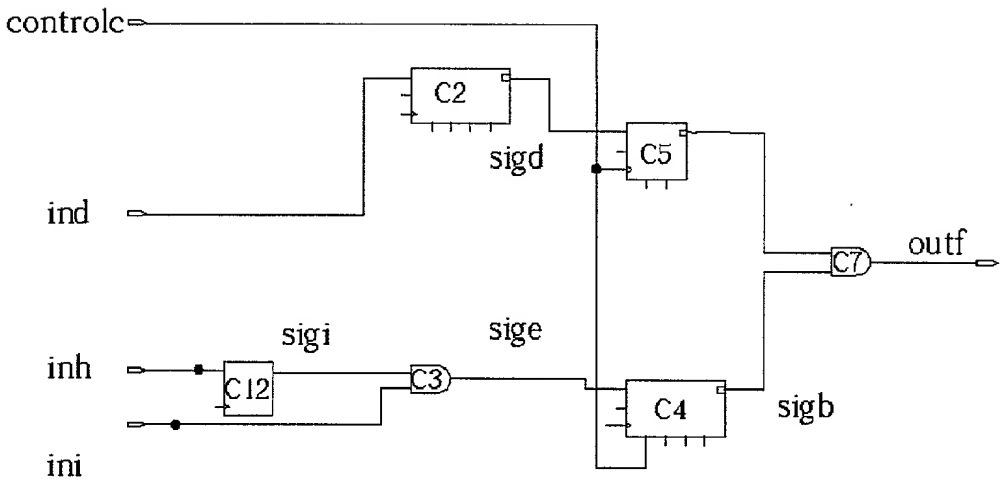

Fig. 17

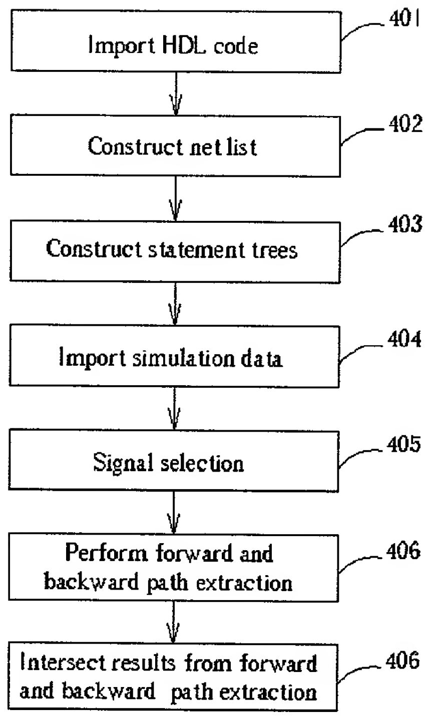
Import HDL code
401
Construct net list
402
Construct statement trees
403
Import simulation data
404
Signal selection
405
Perform forward and backward path extraction
406
Intersect results from forward and backward path extraction
406

Fig. 18

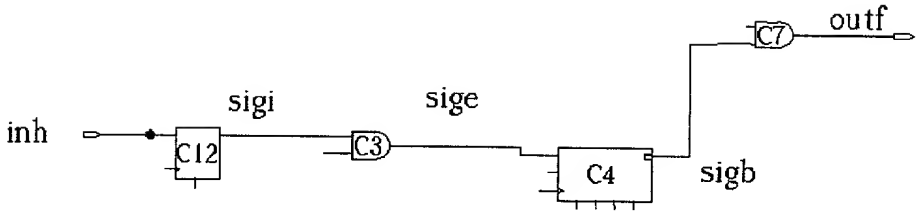

Fig. 19